

Tower Lights Project

Generated by Doxygen 1.8.9.1

Mon May 11 2015 10:19:48

Contents

1	Main Page	1
2	Namespace Index	3
2.1	Namespace List	3
3	Hierarchical Index	5
3.1	Class Hierarchy	5
4	Class Index	7
4.1	Class List	7
5	File Index	9
5.1	File List	9
6	Namespace Documentation	11
6.1	Ui Namespace Reference	11
6.1.1	Detailed Description	11
7	Class Documentation	13
7.1	colorPalette Class Reference	13
7.1.1	Detailed Description	13
7.1.2	Constructor & Destructor Documentation	13
7.1.2.1	colorPalette	13
7.1.2.2	~colorPalette	13
7.1.3	Member Function Documentation	13
7.1.3.1	getColor	14
7.1.4	Member Data Documentation	14
7.1.4.1	count	14
7.2	ColorWheel Class Reference	14
7.2.1	Detailed Description	15
7.2.2	Constructor & Destructor Documentation	15
7.2.2.1	ColorWheel	15
7.2.3	Member Function Documentation	15
7.2.3.1	color	15

7.2.3.2	colorChange	15
7.2.3.3	colorToString	15
7.2.3.4	composeWheel	15
7.2.3.5	drawIndicator	15
7.2.3.6	drawPicker	15
7.2.3.7	drawSquareImage	16
7.2.3.8	drawWheelImage	16
7.2.3.9	hueChanged	16
7.2.3.10	minimumSizeHint	16
7.2.3.11	mouseMoveEvent	16
7.2.3.12	mousePressEvent	16
7.2.3.13	mouseReleaseEvent	16
7.2.3.14	paintEvent	16
7.2.3.15	posColor	16
7.2.3.16	resizeEvent	16
7.2.3.17	setColor	16
7.2.3.18	sizeHint	16
7.2.3.19	svChanged	16
7.2.3.20	updateColor	16
7.2.4	Member Data Documentation	16
7.2.4.1	current	16
7.2.4.2	initSize	16
7.2.4.3	inSquare	16
7.2.4.4	inWheel	16
7.2.4.5	lastPos	16
7.2.4.6	margin	16
7.2.4.7	mouseDown	16
7.2.4.8	squareImage	16
7.2.4.9	squareRegion	16
7.2.4.10	wheel	16
7.2.4.11	wheelImage	16
7.2.4.12	wheelRegion	16
7.2.4.13	wheelWidth	17
7.3	Frame Class Reference	17
7.3.1	Detailed Description	17
7.3.2	Constructor & Destructor Documentation	18
7.3.2.1	Frame	18
7.3.2.2	Frame	18
7.3.3	Member Function Documentation	18
7.3.3.1	applyVector	18

7.3.3.2	FullGridPixel	18
7.3.3.3	getTimeStamp	18
7.3.3.4	setFullGridPixelColor	18
7.3.3.5	setTimeStamp	18
7.3.3.6	setTowerGridPixel	18
7.3.3.7	TowerGridPixel	18
7.3.4	Member Data Documentation	18
7.3.4.1	fullGrid	18
7.3.4.2	timeStamp	18
7.3.4.3	towerGrid	19
7.4	MainWindow Class Reference	19
7.4.1	Detailed Description	22
7.4.2	Constructor & Destructor Documentation	22
7.4.2.1	MainWindow	22
7.4.2.2	~MainWindow	22
7.4.3	Member Function Documentation	22
7.4.3.1	changeCurrentFrame	22
7.4.3.2	closeEvent	22
7.4.3.3	fixPalletBackground	22
7.4.3.4	keyPressEvent	22
7.4.3.5	on_actionAfter_triggered	22
7.4.3.6	on_actionClear_Frame_triggered	22
7.4.3.7	on_actionClose_triggered	22
7.4.3.8	on_actionDelete_triggered	22
7.4.3.9	on_actionExit_triggered	22
7.4.3.10	on_actionExport_triggered	23
7.4.3.11	on_actionNew_triggered	23
7.4.3.12	on_actionOpen_Audio_File_triggered	23
7.4.3.13	on_actionOpen_triggered	23
7.4.3.14	on_actionPreview_Mode_triggered	23
7.4.3.15	on_actionSave_As_triggered	23
7.4.3.16	on_actionSave_triggered	23
7.4.3.17	on_audioDurationChanged	23
7.4.3.18	on_blueSpinBox_valueChanged	23
7.4.3.19	on_currentTime_timeChanged	23
7.4.3.20	on_downButton_clicked	23
7.4.3.21	on_downLeftButton_clicked	23
7.4.3.22	on_downRightButton_clicked	24
7.4.3.23	on_greenSpinBox_valueChanged	24
7.4.3.24	on_leftButton_clicked	24

7.4.3.25	on_mediaSlider_sliderMoved	24
7.4.3.26	on_newFrameButton_clicked	24
7.4.3.27	on_playPauseButton_clicked	24
7.4.3.28	on_previewScrollBar_valueChanged	24
7.4.3.29	on_randomButton_clicked	24
7.4.3.30	on_redSpinBox_valueChanged	24
7.4.3.31	on_rightButton_clicked	24
7.4.3.32	on_SortButton_clicked	24
7.4.3.33	on_stopButton_clicked	24
7.4.3.34	on_upButton_clicked	25
7.4.3.35	on_upLeftButton_clicked	25
7.4.3.36	on_upRightButton_clicked	25
7.4.3.37	saveCurrentFrame	25
7.4.3.38	saveWarning	25
7.4.3.39	setUpMats	25
7.4.3.40	towerPainting	25
7.4.3.41	updateColorSpinBoxes	25
7.4.3.42	updateMainTower	25
7.4.3.43	updateUI	25
7.4.4	Member Data Documentation	25
7.4.4.1	audio	25
7.4.4.2	blue	25
7.4.4.3	colorPallet	26
7.4.4.4	currentMovie	26
7.4.4.5	cWheel	26
7.4.4.6	edited	26
7.4.4.7	fileName	26
7.4.4.8	fullTower	26
7.4.4.9	green	26
7.4.4.10	mainTower	26
7.4.4.11	ping	26
7.4.4.12	previewer	26
7.4.4.13	previewTowers	26
7.4.4.14	red	26
7.4.4.15	saveTimer	27
7.4.4.16	stop	27
7.4.4.17	timer	27
7.4.4.18	ui	27
7.5	Movie Class Reference	27
7.5.1	Detailed Description	28

7.5.2	Constructor & Destructor Documentation	28
7.5.2.1	Movie	28
7.5.3	Member Function Documentation	28
7.5.3.1	deleteCurrentFrame	28
7.5.3.2	getAudioFile	29
7.5.3.3	getCurrentFrame	29
7.5.3.4	getFrame	29
7.5.3.5	getFrameCount	29
7.5.3.6	getFrameFromTime	29
7.5.3.7	getFrameNumber	29
7.5.3.8	getNextFrame	29
7.5.3.9	insertFrame	29
7.5.3.10	newFrame	29
7.5.3.11	newFrame	29
7.5.3.12	printTimeStamps	29
7.5.3.13	reset	29
7.5.3.14	setAudio	30
7.5.3.15	setFile	30
7.5.3.16	setFrame	30
7.5.3.17	setFrameNumber	30
7.5.3.18	setFrameTime	30
7.5.3.19	sortFrames	30
7.5.4	Member Data Documentation	30
7.5.4.1	audioFile	30
7.5.4.2	currentFrameNumber	30
7.5.4.3	frameSequence	30
7.5.4.4	mainFile	30
7.6	Pixel Class Reference	31
7.6.1	Detailed Description	31
7.6.2	Constructor & Destructor Documentation	31
7.6.2.1	Pixel	31
7.6.2.2	Pixel	31
7.6.3	Member Function Documentation	31
7.6.3.1	getColor	31
7.6.3.2	operator=	31
7.6.3.3	setColor	32
7.6.3.4	setColor	32
7.6.4	Member Data Documentation	32
7.6.4.1	color	32
7.7	Previewer Class Reference	32

7.7.1	Detailed Description	33
7.7.2	Constructor & Destructor Documentation	33
7.7.2.1	Previewer	33
7.7.2.2	~Previewer	33
7.7.3	Member Function Documentation	33
7.7.3.1	on_horizontalScrollBar_valueChanged	33
7.7.3.2	on_pushButton_clicked	33
7.7.3.3	setUpMatrix	34
7.7.3.4	showEvent	34
7.7.3.5	updateData	34
7.7.3.6	updateUI	34
7.7.4	Member Data Documentation	34
7.7.4.1	currentAudio	34
7.7.4.2	currentFrameNumber	34
7.7.4.3	currentMovie	34
7.7.4.4	stop	34
7.7.4.5	timer	34
7.7.4.6	tower	34
7.7.4.7	ui	34
7.8	Timer Class Reference	35
7.8.1	Detailed Description	35
7.8.2	Constructor & Destructor Documentation	35
7.8.2.1	Timer	35
7.8.2.2	~Timer	35
7.8.3	Member Function Documentation	35
7.8.3.1	getTime	35
7.8.3.2	start	36
7.8.3.3	start	36
7.8.3.4	stop	36
7.8.4	Member Data Documentation	36
7.8.4.1	offset	36
7.8.4.2	startTime	36
7.8.4.3	timer	36
8	File Documentation	37
8.1	colorpalette.cpp File Reference	37
8.2	colorpalette.h File Reference	37
8.2.1	Detailed Description	37
8.3	colorwheel.cpp File Reference	37
8.3.1	Detailed Description	38

8.4	colorwheel.h File Reference	38
8.4.1	Detailed Description	38
8.5	frame.cpp File Reference	38
8.6	frame.h File Reference	38
8.6.1	Detailed Description	39
8.6.2	Macro Definition Documentation	39
8.6.2.1	FULLGRIDHEIGHT	39
8.6.2.2	FULLGRIDWIDTH	39
8.6.2.3	TOWERHEIGHT	39
8.6.2.4	TOWERWIDTH	39
8.7	main.cpp File Reference	39
8.7.1	Function Documentation	40
8.7.1.1	main	40
8.8	mainwindow.cpp File Reference	40
8.9	mainwindow.h File Reference	40
8.9.1	Detailed Description	40
8.10	mainwindow_util.cpp File Reference	40
8.11	movie.cpp File Reference	41
8.11.1	Function Documentation	41
8.11.1.1	compareFrames	41
8.12	movie.h File Reference	41
8.12.1	Detailed Description	41
8.12.2	Function Documentation	41
8.12.2.1	compareFrames	41
8.13	pixel.cpp File Reference	41
8.14	pixel.h File Reference	42
8.14.1	Detailed Description	42
8.15	previewer.cpp File Reference	42
8.16	previewer.h File Reference	42
8.16.1	Detailed Description	42
8.17	timer.cpp File Reference	43
8.18	timer.h File Reference	43
8.18.1	Detailed Description	43
	Index	45

Chapter 1

Main Page

Title: Tower Animator Software Design

Team Members: 1st Semester:

1. Matthew Brown
2. Ranger Adams
3. Paul Bailey
4. Emeth Thomson

2nd Semester:

1. Matthew Brown
2. Ranger Adams
3. Cameron Simon

This Program was created through the University of Idaho's capstone design class. The use of this program is for the ACM club's tower light show. If you wish to edit the source of this program please do so through the permission of Dr. Rinker at the University of Idaho, Moscow.

Note: This document does not describe the built in Qt functions/classes; only the classes created by the senior design team for this project.

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Ui	11
----------	----

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Frame	17
Movie	27
Pixel	31
QDialog	
Previewer	32
QMainWindow	
MainWindow	19
QObject	
colorPalette	13
QWidget	
ColorWheel	14
Timer	35

Chapter 4

Class Index

4.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

colorPalette	The colorPalette class	13
ColorWheel	The ColorWheel class	14
Frame	The Frame class	17
MainWindow	The MainWindow class	19
Movie	The Movie class	27
Pixel	The Pixel class	31
Previewer	The Previewer class	32
Timer	The Timer class	35

Chapter 5

File Index

5.1 File List

Here is a list of all files with brief descriptions:

colorpalette.cpp	37
colorpalette.h	
This file defines the code side of the colorPalette class. It stores information about default colors	37
colorwheel.cpp	37
colorwheel.h	
This file defines the ColorWheel class	38
frame.cpp	38
frame.h	
This file defines the Frame class as well as the macros FULLGRIDHEIGHT, FULLGRIDWIDTH, TOWERHEIGHT and TOWERWIDTH	38
main.cpp	39
mainwindow.cpp	40
mainwindow.h	
This file defines the MainWindow class of the program	40
mainwindow_util.cpp	40
movie.cpp	41
movie.h	
This file defines the Movie class	41
pixel.cpp	41
pixel.h	
This file defines the Pixel class	42
previewer.cpp	42
previewer.h	
This file defines the Previewer class	42
timer.cpp	43
timer.h	
This file class defines the Timer class	43

Chapter 6

Namespace Documentation

6.1 Ui Namespace Reference

6.1.1 Detailed Description

The [Ui](#) Namespace is used to by the Qt envoinrment. Its used to be able gather widget data from GUI developed from XML forms.

Chapter 7

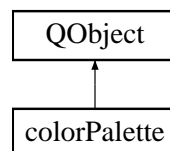
Class Documentation

7.1 colorPalette Class Reference

The `colorPalette` class.

```
#include <colorpalette.h>
```

Inheritance diagram for `colorPalette`:



Public Member Functions

- `colorPalette()`
- `~colorPalette()`
- `Q_INVOKABLE QString getColor()`

Private Attributes

- `int count`

7.1.1 Detailed Description

The `colorPalette` class.

This class serves as a helper class to the Color Palette QML code. It keeps track of which color is currently being produced and return a color in accordance.

7.1.2 Constructor & Destructor Documentation

7.1.2.1 `colorPalette::colorPalette()`

7.1.2.2 `colorPalette::~~colorPalette()`

7.1.3 Member Function Documentation

7.1.3.1 QString colorPalette::getColor ()

7.1.4 Member Data Documentation

7.1.4.1 int colorPalette::count [private]

keeps track of the current color

The documentation for this class was generated from the following files:

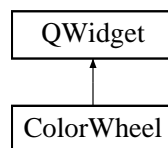
- [colorpalette.h](#)
- [colorpalette.cpp](#)

7.2 ColorWheel Class Reference

The [ColorWheel](#) class.

```
#include <colorwheel.h>
```

Inheritance diagram for ColorWheel:



Public Slots

- void [setColor](#) (const QColor &[color](#))
- void [updateColor](#) (const QColor &[color](#))

Signals

- void [colorChange](#) (const QColor &[color](#))

Public Member Functions

- [ColorWheel](#) (QWidget *parent=0)
- virtual QSize [sizeHint](#) () const
- virtual QSize [minimumSizeHint](#) () const
- QColor [color](#) ()
- Q_INVOKABLE QString [colorToString](#) ()

Protected Member Functions

- void [mousePressEvent](#) (QMouseEvent *event)
- void [mouseMoveEvent](#) (QMouseEvent *event)
- void [mouseReleaseEvent](#) (QMouseEvent *)
- void [resizeEvent](#) (QResizeEvent *event)
- void [paintEvent](#) (QPaintEvent *)

Private Slots

- void [hueChanged](#) (const int &hue)
- void [svChanged](#) (const QColor &newcolor)

Private Member Functions

- QColor [posColor](#) (const QPoint &point)
- void [drawWheelImage](#) (const QSize &newSize)
- void [drawIndicator](#) (const int &hue)
- void [drawPicker](#) (const QColor &color)
- void [drawSquareImage](#) (const int &hue)
- void [composeWheel](#) ()

Private Attributes

- QSize [initSize](#)
- QImage [wheelImage](#)
- QImage [squareImage](#)
- QPixmap [wheel](#)
- bool [mouseDown](#)
- QPoint [lastPos](#)
- int [margin](#)
- int [wheelWidth](#)
- QRegion [wheelRegion](#)
- QRegion [squareRegion](#)
- QColor [current](#)
- bool [inWheel](#)
- bool [inSquare](#)

7.2.1 Detailed Description

The [ColorWheel](#) class.

7.2.2 Constructor & Destructor Documentation

7.2.2.1 [ColorWheel::ColorWheel \(QWidget * *parent* = 0 \)](#) [\[explicit\]](#)

7.2.3 Member Function Documentation

7.2.3.1 [QColor ColorWheel::color \(\)](#)

7.2.3.2 [void ColorWheel::colorChange \(const QColor & *color* \)](#) [\[signal\]](#)

7.2.3.3 [QString ColorWheel::colorToString \(\)](#)

7.2.3.4 [void ColorWheel::composeWheel \(\)](#) [\[private\]](#)

7.2.3.5 [void ColorWheel::drawIndicator \(const int & *hue* \)](#) [\[private\]](#)

7.2.3.6 [void ColorWheel::drawPicker \(const QColor & *color* \)](#) [\[private\]](#)

- 7.2.3.7 void ColorWheel::drawSquareImage (const int & *hue*) [private]
- 7.2.3.8 void ColorWheel::drawWheelImage (const QSize & *newSize*) [private]
- 7.2.3.9 void ColorWheel::hueChanged (const int & *hue*) [private],[slot]
- 7.2.3.10 QSize ColorWheel::minimumSizeHint () const [virtual]
- 7.2.3.11 void ColorWheel::mouseMoveEvent (QMouseEvent * *event*) [protected]
- 7.2.3.12 void ColorWheel::mousePressEvent (QMouseEvent * *event*) [protected]
- 7.2.3.13 void ColorWheel::mouseReleaseEvent (QMouseEvent *) [protected]
- 7.2.3.14 void ColorWheel::paintEvent (QPaintEvent *) [protected]
- 7.2.3.15 QColor ColorWheel::posColor (const QPoint & *point*) [private]
- 7.2.3.16 void ColorWheel::resizeEvent (QResizeEvent * *event*) [protected]
- 7.2.3.17 void ColorWheel::setColor (const QColor & *color*) [slot]
- 7.2.3.18 QSize ColorWheel::sizeHint () const [virtual]
- 7.2.3.19 void ColorWheel::svChanged (const QColor & *newcolor*) [private],[slot]
- 7.2.3.20 void ColorWheel::updateColor (const QColor & *color*) [slot]

7.2.4 Member Data Documentation

- 7.2.4.1 QColor ColorWheel::current [private]
- 7.2.4.2 QSize ColorWheel::initSize [private]
- 7.2.4.3 bool ColorWheel::inSquare [private]
- 7.2.4.4 bool ColorWheel::inWheel [private]
- 7.2.4.5 QPoint ColorWheel::lastPos [private]
- 7.2.4.6 int ColorWheel::margin [private]
- 7.2.4.7 bool ColorWheel::mouseDown [private]
- 7.2.4.8 QImage ColorWheel::squareImage [private]
- 7.2.4.9 QRegion ColorWheel::squareRegion [private]
- 7.2.4.10 QPixmap ColorWheel::wheel [private]
- 7.2.4.11 QImage ColorWheel::wheelImage [private]
- 7.2.4.12 QRegion ColorWheel::wheelRegion [private]

7.2.4.13 `int ColorWheel::wheelWidth` `[private]`

The documentation for this class was generated from the following files:

- [colorwheel.h](#)
- [colorwheel.cpp](#)

7.3 Frame Class Reference

The `Frame` class.

```
#include <frame.h>
```

Public Member Functions

- `Frame ()`
Constructor.
- `Frame (const Frame &original)`
Copy Constructor.
- `void setTimeStamp (qint64 time)`
Setter for the timeStamp private memeber.
- `qint64 getTimeStamp ()`
Getter for the timeStamp private member.
- `void setFullGridPixelColor (int x, int y, QColor c)`
Sets a pixel's color based on the full 12x20 grid.
- `void setTowerGridPixel (int x, int y, QColor c)`
Sets a pixel's color based on the 4x10.
- `Pixel * FullGridPixel (int x, int y)`
Gets a pixel pointer based on the 12x20 coordinate system.
- `Pixel * TowerGridPixel (int x, int y)`
Gets a pixel pointer based on the 4x10 coordinate system.
- `void applyVector (int x, int y)`
Shifts the pixels in a frame by x y.

Private Attributes

- `Pixel * fullGrid [20][12]`
A matrix of pointers that contains pixel data.
- `Pixel * towerGrid [10][4]`
- `qint64 timeStamp`
Time information for when the fram should appear.

7.3.1 Detailed Description

The `Frame` class.

The `Frame` class contains the data that each frame uses. This data includes the 2D color matrices and the time information for when the frame is appear.

7.3.2 Constructor & Destructor Documentation

7.3.2.1 `Frame::Frame ()`

Constructor.

7.3.2.2 `Frame::Frame (const Frame & original)`

Copy Constructor.

7.3.3 Member Function Documentation

7.3.3.1 `void Frame::applyVector (int x, int y)`

Shifts the pixels in a frame by *x y*.

7.3.3.2 `Pixel * Frame::FullGridPixel (int x, int y)`

Gets a pixel pointer based on the 12x20 coordinate system.

7.3.3.3 `qint64 Frame::getTimeStamp ()`

Getter for the `timeStamp` private member.

7.3.3.4 `void Frame::setFullGridPixelColor (int x, int y, QColor c)`

Sets a pixel's color based on the full 12x20 grid.

7.3.3.5 `void Frame::setTimeStamp (qint64 time)`

Setter for the `timeStamp` private memeber.

7.3.3.6 `void Frame::setTowerGridPixel (int x, int y, QColor c)`

Sets a pixel's color based on the 4x10.

7.3.3.7 `Pixel * Frame::TowerGridPixel (int x, int y)`

Gets a pixel pointer based on the 4x10 coordinate system.

7.3.4 Member Data Documentation

7.3.4.1 `Pixel* Frame::fullGrid[20][12] [private]`

A matrix of pointers that contains pixel data.

7.3.4.2 `qint64 Frame::timeStamp [private]`

Time information for when the fram should appear.

7.3.4.3 Pixel* Frame::towerGrid[10][4] [private]

A subset of fullGrid. Used to point to data only in the main tower. No new data is created this just points to elements allocated in fullGrid

The documentation for this class was generated from the following files:

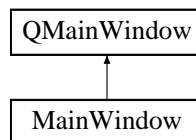
- [frame.h](#)
- [frame.cpp](#)

7.4 MainWindow Class Reference

The [MainWindow](#) class.

```
#include <mainwindow.h>
```

Inheritance diagram for MainWindow:



Public Member Functions

- [MainWindow](#) (QWidget *parent=0)
- [~MainWindow](#) ()

Private Slots

- void [on_actionExit_triggered](#) ()
This function is called when file->exit is triggered.
- void [on_actionNew_triggered](#) ()
This function is called when file->new is triggered.
- void [on_actionExport_triggered](#) ()
This function is called when file->Export is triggered.
- void [on_actionSave_As_triggered](#) ()
This function is called when file->SaveAs is triggered.
- void [on_actionSave_triggered](#) ()
This function is called when file->Save is triggered.
- void [on_redSpinBox_valueChanged](#) (int arg1)
This function is called when the red color value is changed.
- void [on_blueSpinBox_valueChanged](#) (int arg1)
This function is called when the blue color value is changed.
- void [on_greenSpinBox_valueChanged](#) (int arg1)
This function is called when the green color value is changed.
- void [updateColorSpinBoxes](#) (QColor)
This function is called when the color changed signal in the color wheel class is triggered.
- void [towerPainting](#) ()
This function is called when the user changes the tower colors.
- void [on_actionOpen_Audio_File_triggered](#) ()

- This function is called when file-> Open Audio File is triggered.*

 - void [on_actionOpen_triggered](#) ()
- This function is called when file->Open is triggered.*

 - void [on_newFrameButton_clicked](#) ()
- This function is called when the newFrame button is triggered.*

 - void [on_previewScrollBar_valueChanged](#) (int value)
- This function is called when Scroll bar changes values.*

 - void [on_actionAfter_triggered](#) ()
- This function is called when edit->insert->after is triggered.*

 - void [on_currentTime_timeChanged](#) (const QTime &time)
- This function is called when the current time value is changed.*

 - void [on_playPauseButton_clicked](#) ()
- This function is called when the play/pause button is clicked.*

 - void [on_stopButton_clicked](#) ()
- This function is called when the stop button is clicked.*

 - void [on_upButton_clicked](#) ()
- This function is called when the up arrow button is clicked.*

 - void [on_randomButton_clicked](#) ()
- This function is called when the random button is clicked.*

 - void [on_downButton_clicked](#) ()
- This function is called when the down arrow button is clicked.*

 - void [on_leftButton_clicked](#) ()
- This function is called when the left arrow button is clicked.*

 - void [on_rightButton_clicked](#) ()
- This function is called when the right arrow button is clicked.*

 - void [on_downLeftButton_clicked](#) ()
- This function is called when the down left arrow button is clicked.*

 - void [on_upLeftButton_clicked](#) ()
- This function is called when the up left arrow button is clicked.*

 - void [on_upRightButton_clicked](#) ()
- This function is called when the up right arrow button is clicked.*

 - void [on_downRightButton_clicked](#) ()
- This function is called when the down right arrow button is clicked.*

 - void [keyPressEvent](#) (QKeyEvent *event)
- This function is called when playback->previewmode is clicked.*

 - void [on_actionPreview_Mode_triggered](#) ()
- This function is called when file->close is triggered.*

 - void [on_actionClose_triggered](#) ()
- This function is called when the media slider is moved.*

 - void [on_mediaSlider_sliderMoved](#) (int position)
- This function is called when the audio's length changes like when the file changes.*

 - void [on_audioDurationChanged](#) (qint64 duration)
- This function is called when the sort button is clicked.*

 - void [on_SortButton_clicked](#) ()
- This function is called when the sort button is clicked.*

 - void [on_actionClear_Frame_triggered](#) ()
- This function is called when the sort button is clicked.*

 - void [on_actionDelete_triggered](#) ()

Private Member Functions

- void `fixPalletBackground` ()
helper function used to give the color pallet the correct background
- void `saveCurrentFrame` ()
helper function used to save the current tower's data to a frame
- void `updateUI` ()
helper function used to update the UI based on new data/signals
- void `setUpMats` ()
helper function used to set up the QQuickItem matrices
- void `changeCurrentFrame` (int value)
helper function called when the scroll bar changes value.
- void `updateMainTower` ()
helper function used to update the graphical tower to the data stored
- void `closeEvent` (QCloseEvent *event)
Called when the x is clicked on the main window.

Private Attributes

- Ui::MainWindow * `ui`
The ui is used to interface with the xml generated GUI.
- `Previewer` `previewer`
This is the previewer window.
- `ColorWheel` * `cWheel`
This is the colorwheel object.
- `QSpinBox` * `red`
This is a spinbox that holds the red value of the current color.
- `QSpinBox` * `blue`
This is a spinbox that holds the blue value of the current color.
- `QSpinBox` * `green`
This is a spinbox that holds the green value of the current color.
- `QQuickItem` * `previewTowers` [7][10][4]
This matrix points to the quick items that make up the preview section.
- `QQuickItem` * `fullTower` [20][12]
This matrix points to the quick items that make up the full tower area.
- `QQuickItem` * `mainTower` [10][4]
This matrix points to the quick items that make up the main tower area.
- `QQuickItem` * `colorPallet` [2][8]
This matrix points to the quick items that make up the color pallet.
- bool `ping`
This boolean is used to control the coordination of the color selectors.
- bool `stop`
This boolean is used to control whether the GUI is in playback mode or not.
- bool `edited`
This boolean signifies whether or not the tower has been edited since last save.
- `Movie` * `currentMovie`
This is the main move object used to contain the frame sequence data.
- `QMediaPlayer` * `audio`
- `Timer` `timer`
This timer is used to control the frame sequence timing when no audio is present.
- `QTimer` * `saveTimer`
This timer is used to control when the program should auto save next.
- `QString` `fileName`
Contains the fileName of the current movie.

7.4.1 Detailed Description

The [MainWindow](#) class.

The [MainWindow](#) class defines the GUI of the program. Its used to connect the buttons to the the C++ functions. QT uses its own handles for things like drawing the GUI and processing events so all that the programmer has to do is define high level behavior. This class inherits the QMainWindow class that is defined by QT. There are these new kind of member functions called slot functions. These functions are connected to parts of the GUI and are predefined in the QT framework when the interface object is created. The name of the function is required and cannot be changed without effecting the behavior of the program. This class also hold data such as matrices that are design to make addressing the on screen tower's easier. The class holds other members that will be defined later on.

7.4.2 Constructor & Destructor Documentation

7.4.2.1 `MainWindow::MainWindow (QWidget * parent = 0)` `[explicit]`

7.4.2.2 `MainWindow::~~MainWindow ()`

7.4.3 Member Function Documentation

7.4.3.1 `void MainWindow::changeCurrentFrame (int value)` `[private]`

helper function called when the scroll bar changes value.

7.4.3.2 `void MainWindow::closeEvent (QCloseEvent * event)` `[private]`

Called when the x is clicked on the main window.

7.4.3.3 `void MainWindow::fixPalletBackground ()` `[private]`

helper function used to give the color pallet the correct background

7.4.3.4 `void MainWindow::keyPressEvent (QKeyEvent * event)` `[private],[slot]`

This function is called when any keyboard button is pressed. A switch statement is used to filter exactly which button was pressed.

7.4.3.5 `void MainWindow::on_actionAfter_triggered ()` `[private],[slot]`

This function is called when edit->insert->after is triggered.

7.4.3.6 `void MainWindow::on_actionClear_Frame_triggered ()` `[private],[slot]`

7.4.3.7 `void MainWindow::on_actionClose_triggered ()` `[private],[slot]`

This function is called when file->close is triggered.

7.4.3.8 `void MainWindow::on_actionDelete_triggered ()` `[private],[slot]`

7.4.3.9 `void MainWindow::on_actionExit_triggered ()` `[private],[slot]`

This function is called when file->exit is triggered.

7.4.3.10 `void MainWindow::on_actionExport_triggered () [private],[slot]`

This function is called when file->Export is triggered.

7.4.3.11 `void MainWindow::on_actionNew_triggered () [private],[slot]`

This function is called when file->new is triggered.

7.4.3.12 `void MainWindow::on_actionOpen_Audio_File_triggered () [private],[slot]`

This function is called when file-> Open Audio File is triggered.

7.4.3.13 `void MainWindow::on_actionOpen_triggered () [private],[slot]`

This function is called when file->Open is triggered.

7.4.3.14 `void MainWindow::on_actionPreview_Mode_triggered () [private],[slot]`

This function is called when playback->previewmode is clicked.

7.4.3.15 `void MainWindow::on_actionSave_As_triggered () [private],[slot]`

This function is called when file->SaveAs is triggered.

7.4.3.16 `void MainWindow::on_actionSave_triggered () [private],[slot]`

This function is called when file->Save is triggered.

7.4.3.17 `void MainWindow::on_audioDurationChanged (qint64 duration) [private],[slot]`

This function is called when the audio's length changes like when the file changes.

7.4.3.18 `void MainWindow::on_blueSpinBox_valueChanged (int arg1) [private],[slot]`

This function is called when the blue color value is changed.

7.4.3.19 `void MainWindow::on_currentTime_timeChanged (const QTime & time) [private],[slot]`

This function is called when the current time value is changed.

7.4.3.20 `void MainWindow::on_downButton_clicked () [private],[slot]`

This function is called when the down arrow button is clicked.

7.4.3.21 `void MainWindow::on_downLeftButton_clicked () [private],[slot]`

This function is called when the down left arrow button is clicked.

7.4.3.22 `void MainWindow::on_downRightButton_clicked () [private],[slot]`

This function is called when the down right arrow button is clicked.

7.4.3.23 `void MainWindow::on_greenSpinBox_valueChanged (int arg1) [private],[slot]`

This function is called when the green color value is changed.

7.4.3.24 `void MainWindow::on_leftButton_clicked () [private],[slot]`

This function is called when the left arrow button is clicked.

7.4.3.25 `void MainWindow::on_mediaSlider_sliderMoved (int position) [private],[slot]`

This function is called when the media slider is moved.

7.4.3.26 `void MainWindow::on_newFrameButton_clicked () [private],[slot]`

This function is called when the newFrame button is triggered.

7.4.3.27 `void MainWindow::on_playPauseButton_clicked () [private],[slot]`

This function is called when the play/pause button is clicked.

7.4.3.28 `void MainWindow::on_previewScrollBar_valueChanged (int value) [private],[slot]`

This function is called when Scroll bar changes values.

7.4.3.29 `void MainWindow::on_randomButton_clicked () [private],[slot]`

This function is called when the random button is clicked.

7.4.3.30 `void MainWindow::on_redSpinBox_valueChanged (int arg1) [private],[slot]`

This function is called when the red color value is changed.

7.4.3.31 `void MainWindow::on_rightButton_clicked () [private],[slot]`

This function is called when the right arrow button is clicked.

7.4.3.32 `void MainWindow::on_SortButton_clicked () [private],[slot]`

This function is called when the sort button is clicked.

7.4.3.33 `void MainWindow::on_stopButton_clicked () [private],[slot]`

This function is called when the stop button is clicked.

7.4.3.34 `void MainWindow::on_upButton_clicked () [private],[slot]`

This function is called when the up arrow button is clicked.

7.4.3.35 `void MainWindow::on_upLeftButton_clicked () [private],[slot]`

This function is called when the up left arrow button is clicked.

7.4.3.36 `void MainWindow::on_upRightButton_clicked () [private],[slot]`

This function is called when the up right arrow button is clicked.

7.4.3.37 `void MainWindow::saveCurrentFrame () [private]`

helper function used to save the current tower's data to a frame

7.4.3.38 `void MainWindow::saveWarning () [private],[slot]`

This function is called when a timer runs out. This timer controls the autosave feature of the program.

7.4.3.39 `void MainWindow::setUpMats () [private]`

helper function used to set up the QQuickItem matrices

7.4.3.40 `void MainWindow::towerPainting () [private],[slot]`

This function is called when the user changes the tower colors.

7.4.3.41 `void MainWindow::updateColorSpinBoxes (QColor newColor) [private],[slot]`

This function is called when the color changed signal in the color wheel class is triggered.

7.4.3.42 `void MainWindow::updateMainTower () [private]`

helper function used to update the graphical tower to the data stored

7.4.3.43 `void MainWindow::updateUI () [private]`

helper function used to update the UI based on new data/signals

7.4.4 Member Data Documentation

7.4.4.1 `QMediaPlayer* MainWindow::audio [private]`

audio is used to load and contain the music of the program. This class is a little tricky to use but documentation can be found at <http://doc.qt.io/qt-5/qmediaplayer.html>

7.4.4.2 `QSpinBox* MainWindow::blue [private]`

This is a spinbox that holds the blue value of the current color.

7.4.4.3 `QQuickItem* MainWindow::colorPallet[2][8]` [private]

This matrix points to the quick items that make up the color pallet.

7.4.4.4 `Movie* MainWindow::currentMovie` [private]

This is the main move object used to contain the frame sequence data.

7.4.4.5 `ColorWheel* MainWindow::cWheel` [private]

This is the colorwheel object.

7.4.4.6 `bool MainWindow::edited` [private]

This boolean signifies whether or not the tower has been edited since last save.

7.4.4.7 `QString MainWindow::fileName` [private]

Contains the fileName of the current movie.

7.4.4.8 `QQuickItem* MainWindow::fullTower[20][12]` [private]

This matrix points to the quick items that make up the full tower area.

7.4.4.9 `QSpinBox* MainWindow::green` [private]

This is a spinbox that holds the green value of the current color.

7.4.4.10 `QQuickItem* MainWindow::mainTower[10][4]` [private]

This matrix points to the quick items that make up the main tower area.

7.4.4.11 `bool MainWindow::ping` [private]

This boolean is used to control the coordination of the color selectors.

7.4.4.12 `Previewer MainWindow::previewer` [private]

This is the previewer window.

7.4.4.13 `QQuickItem* MainWindow::previewTowers[7][10][4]` [private]

This matrix points to the quick items that make up the preview section.

7.4.4.14 `QSpinBox* MainWindow::red` [private]

This is a spinbox that holds the red value of the current color.

7.4.4.15 `QTimer* MainWindow::saveTimer` [private]

This timer is used to control when the program should auto save next.

7.4.4.16 `bool MainWindow::stop` [private]

This boolean is used to control whether the GUI is in playback mode or not.

7.4.4.17 `Timer MainWindow::timer` [private]

This timer is used to control the frame sequence timing when no audio is present.

7.4.4.18 `Ui::MainWindow* MainWindow::ui` [private]

The ui is used to interface with the xml generated GUI.

The documentation for this class was generated from the following files:

- [mainwindow.h](#)
- [mainwindow.cpp](#)
- [mainwindow_util.cpp](#)

7.5 Movie Class Reference

The [Movie](#) class.

```
#include <movie.h>
```

Public Member Functions

- [Movie](#) ()
Constructor.
- void [setAudio](#) (QUrl url)
Sets the Audio file's URL.
- void [setFile](#) (QUrl url)
Sets the tan file's URL.
- QUrl [getAudioFile](#) ()
Returns the audio files URL.
- int [getFrameCount](#) ()
Return total number of frames.
- void [newFrame](#) ()
Creates a new frame at the end of the list.
- void [newFrame](#) (int index)
Creates a new frame at position index in the list.
- void [insertFrame](#) (int index, [Frame](#) *f)
Inserts frame f at position index.
- void [setFrame](#) (int index, [Frame](#) *f)
Replaces the fram a position index with frame f.
- void [setFrameTime](#) (int index, qint64 time)
Sets the time of the frame at posistion index to time.
- int [getFrameNumber](#) ()

- return the current frame number*
- [Frame * getNextFrame \(\)](#)
returns a pointer to the frame after the current frame
- [Frame * getFrame \(int i\)](#)
retrns a pointer to the frame at i
- void [reset \(\)](#)
Sets the current frame back to the beggining.
- void [setFrameNumber \(int x\)](#)
Sets the current frame number to x.
- [Frame * getCurrentFrame \(\)](#)
Returns a pointer to the current frame.
- void [sortFrames \(\)](#)
Sorts the frames.
- void [printTimeStamps \(\)](#)
prints the frames timestamps used for debugging
- int [getFrameFromTime \(int time\)](#)
returns the index of the frame at time time
- void [deleteCurrentFrame \(\)](#)
Called to delete the current frame.

Private Attributes

- [QUrl mainFile](#)
hold the URL of the tan file
- [QUrl audioFile](#)
hold the URL of the audio file
- [QList< Frame * > * frameSequence](#)
A list of the frames in the movie.
- int [currentFrameNumber](#)
Contains the index of the current frame.

7.5.1 Detailed Description

The [Movie](#) class.

This class is the main class that contains the back-end data used by the creator. It contains the frame sequence as well as indexing and filename info for both the .tan2 and the audio files.

7.5.2 Constructor & Destructor Documentation

7.5.2.1 [Movie::Movie \(\)](#)

Constructor.

7.5.3 Member Function Documentation

7.5.3.1 [void Movie::deleteCurrentFrame \(\)](#)

Called to delete the current frame.

7.5.3.2 `QString Movie::getAudioFile ()`

Returns the audio files URL.

7.5.3.3 `Frame * Movie::getCurrentFrame ()`

Returns a pointer to the current frame.

7.5.3.4 `Frame * Movie::getFrame (int i)`

retrns a pointer to the frame at i

7.5.3.5 `int Movie::getFrameCount ()`

Return total number of frames.

7.5.3.6 `int Movie::getFrameFromTime (int time)`

returns the index of the frame at time time

7.5.3.7 `int Movie::getFrameNumber ()`

return the current frame number

7.5.3.8 `Frame * Movie::getNextFrame ()`

returns a pointer to the frame after the current frame

7.5.3.9 `void Movie::insertFrame (int index, Frame * f)`

Inserts frame f at position index.

7.5.3.10 `void Movie::newFrame ()`

Creates a new frame at the end of the list.

7.5.3.11 `void Movie::newFrame (int index)`

Creates a new frame at position index in the list.

7.5.3.12 `void Movie::printTimeStamps ()`

prints the frames timestamps used for debugging

7.5.3.13 `void Movie::reset ()`

Sets the current frame back to the beggining.

7.5.3.14 void Movie::setAudio (*QUrl url*)

Sets the Audio file's URL.

7.5.3.15 void Movie::setFile (*QUrl url*)

Sets the tan file's URL.

7.5.3.16 void Movie::setFrame (*int index*, *Frame * f*)

Replaces the fram a position index with frame f.

7.5.3.17 void Movie::setFrameNumber (*int x*)

Sets the current frame number to x.

7.5.3.18 void Movie::setFrameTime (*int index*, *qint64 time*)

Sets the time of the frame at posistion index to time.

7.5.3.19 void Movie::sortFrames ()

Sorts the frames.

7.5.4 Member Data Documentation

7.5.4.1 *QUrl* Movie::audioFile [private]

hold the URL of the audio file

7.5.4.2 *int* Movie::currentFrameNumber [private]

Contains the index of the current frame.

7.5.4.3 *QList<Frame*>** Movie::frameSequence [private]

A list of the frames in the movie.

7.5.4.4 *QUrl* Movie::mainFile [private]

hold the URL of the tan file

The documentation for this class was generated from the following files:

- [movie.h](#)
- [movie.cpp](#)

7.6 Pixel Class Reference

The [Pixel](#) class.

```
#include <pixel.h>
```

Public Member Functions

- [Pixel](#) ()
Constructs a Grey [Pixel](#).
- [Pixel](#) (QColor c)
Constructs a [Pixel](#) Based on the passed QColor value.
- void [setColor](#) (int r, int g, int b)
Sets the pixels color based on rgb colors.
- void [setColor](#) (QColor c)
Sets the pixel color based on a QColor value.
- QColor [getColor](#) ()
Returns a QColor value of the current color.
- void [operator=](#) (const [Pixel](#) &other)
Allows = to be used to set pixels.

Private Attributes

- QColor [color](#)
Holds the color of the pixel.

7.6.1 Detailed Description

The [Pixel](#) class.

This class contains the color data of the windows for the data structure that stores the frame sequence.

7.6.2 Constructor & Destructor Documentation

7.6.2.1 [Pixel::Pixel](#) ()

Constructs a Grey [Pixel](#).

7.6.2.2 [Pixel::Pixel](#) (QColor c)

Constructs a [Pixel](#) Based on the passed QColor value.

7.6.3 Member Function Documentation

7.6.3.1 QColor [Pixel::getColor](#) ()

Returns a QColor value of the current color.

7.6.3.2 void [Pixel::operator=](#) (const [Pixel](#) & other)

Allows = to be used to set pixels.

7.6.3.3 void Pixel::setColor (int *r*, int *g*, int *b*)

Sets the pixels color based on rgb colors.

7.6.3.4 void Pixel::setColor (QColor *c*)

Sets the pixel color based on a QColor value.

7.6.4 Member Data Documentation

7.6.4.1 QColor Pixel::color [private]

Holds the color of the pixel.

The documentation for this class was generated from the following files:

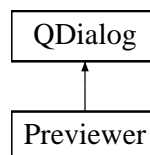
- [pixel.h](#)
- [pixel.cpp](#)

7.7 Previewer Class Reference

The [Previewer](#) class.

```
#include <previewer.h>
```

Inheritance diagram for Previewer:



Public Member Functions

- [Previewer](#) (QWidget *parent=0)
Constructor.
- [~Previewer](#) ()
Destructor.
- void [updateData](#) (Movie *movie, QMediaPlayer *audio)
Updates the pointer for the move and auido data.

Private Slots

- void [on_pushButton_clicked](#) ()
slot function connected to the play button on the GUI
- void [on_horizontalScrollBar_valueChanged](#) (int value)
slot function that is called when the scrollbar's value changes
- void [showEvent](#) (QShowEvent *event)
slot function called when the window is shown

Private Member Functions

- void `setUpMatrix` ()
Sets up the tower matrix.
- void `updateUI` ()
Updates and redraws the GUI.

Private Attributes

- `Ui::Previewer` * `ui`
used to be able to address the content in the GUI created by the XML forms
- `Movie` * `currentMovie`
pointer that points to the movie data in mainWindow
- `QMediaPlayer` * `currentAudio`
points to the audio data in mainWindow
- `QQuickItem` * `tower` [10][4]
- `Timer` * `timer`
The previewer uses its own timer so it doesn't interfere with the mainWindow's.
- int `currentFrameNumber`
Contains the current frame's sequence number.
- bool `stop`
Used in playing to determine whether or not its playing.

7.7.1 Detailed Description

The `Previewer` class.

This class is used to define the look and behavior of the previewer window.

7.7.2 Constructor & Destructor Documentation

7.7.2.1 `Previewer::Previewer (QWidget * parent = 0) [explicit]`

Constructor.

7.7.2.2 `Previewer::~~Previewer ()`

Destructor.

7.7.3 Member Function Documentation

7.7.3.1 `void Previewer::on_horizontalScrollBar_valueChanged (int value) [private], [slot]`

slot function that is called when the scrollbar's value changes

7.7.3.2 `void Previewer::on_pushButton_clicked () [private], [slot]`

slot function connected to the play button on the GUI

7.7.3.3 void Previewer::setUpMatrix () [private]

Sets up the tower matrix.

7.7.3.4 void Previewer::showEvent (QShowEvent * *event*) [private],[slot]

slot function called when the window is shown

7.7.3.5 void Previewer::updateData (Movie * *movie*, QMediaPlayer * *audio*)

Updates the pointer for the move and audio data.

7.7.3.6 void Previewer::updateUI () [private]

Updates and redraws the GUI.

7.7.4 Member Data Documentation

7.7.4.1 QMediaPlayer* Previewer::currentAudio [private]

points to the audio data in mainWindow

7.7.4.2 int Previewer::currentFrameNumber [private]

Contains the current frame's sequence number.

7.7.4.3 Movie* Previewer::currentMovie [private]

pointer that points to the movie data in mainWindow

7.7.4.4 bool Previewer::stop [private]

Used in playing to determine whether or not its playing.

7.7.4.5 Timer* Previewer::timer [private]

The previewer uses its own timer so it doesn't interfere with the mainWindow's.

7.7.4.6 QQuickItem* Previewer::tower[10][4] [private]

matrix of the tower windows on the GUI used for easy access

7.7.4.7 Ui::Previewer* Previewer::ui [private]

used to be able to address the content in the GUI created by the XML forms

The documentation for this class was generated from the following files:

- [previewer.h](#)
- [previewer.cpp](#)

7.8 Timer Class Reference

The [Timer](#) class.

```
#include <timer.h>
```

Public Member Functions

- [Timer](#) ()
Constructor.
- [~Timer](#) ()
Destructor.
- void [start](#) ()
Starts the timer at 0.
- void [start](#) (qint64 [offset](#))
Starts the timer at offset.
- void [stop](#) ()
Stops the timer.
- qint64 [getTime](#) ()
Returns the current time; 0 if not started.

Private Attributes

- QTime [timer](#)
the timer class from Qt
- qint64 [startTime](#)
startTime used for calculating time elapsed
- qint64 [offset](#)
offset used in start(offset)

7.8.1 Detailed Description

The [Timer](#) class.

This class is used in the case that there is no audio file. It is used as a timer to keep the frames in sync with a time format. This classes uses the QTime library to extract time data from the OS.

7.8.2 Constructor & Destructor Documentation

7.8.2.1 [Timer::Timer](#) ()

Constructor.

7.8.2.2 [Timer::~~Timer](#) ()

Destructor.

7.8.3 Member Function Documentation

7.8.3.1 [qint64 Timer::getTime](#) ()

Returns the current time; 0 if not started.

7.8.3.2 void Timer::start ()

Starts the timer at 0.

7.8.3.3 void Timer::start (qint64 *offset*)

Starts the timer at offset.

7.8.3.4 void Timer::stop ()

Stops the timer.

7.8.4 Member Data Documentation

7.8.4.1 qint64 Timer::offset [private]

offset used in start(offset)

7.8.4.2 qint64 Timer::startTime [private]

startTime used for calculating time elapsed

7.8.4.3 QTime Timer::timer [private]

the timer class from Qt

The documentation for this class was generated from the following files:

- [timer.h](#)
- [timer.cpp](#)

Chapter 8

File Documentation

8.1 colorpalette.cpp File Reference

```
#include "colorpalette.h"
```

8.2 colorpalette.h File Reference

This file defines the code side of the [colorPalette](#) class. It stores information about default colors.

```
#include <QObject>
#include <QString>
```

Classes

- class [colorPalette](#)

The [colorPalette](#) class.

8.2.1 Detailed Description

This file defines the code side of the [colorPalette](#) class. It stores information about default colors.

8.3 colorwheel.cpp File Reference

```
#include "colorwheel.h"
#include <QPainter>
#include <QResizeEvent>
#include <QStyleOption>
#include <QtCore/qmath.h>
#include <QDebug>
#include <iostream>
```

8.3.1 Detailed Description

`colorwheel.h` and `colorwheel.cpp` are adapted from a github repository accesible at <https://github.com/liuyanghejerry/Qt-Plus/tree/master/develop/ColorWheel>

The author username is liuyanghejerry and he has given a universal concent to "use it in any purpose." This is locatable at: <http://qt-project.org/forums/viewthread/19757>

8.4 colorwheel.h File Reference

This file defines the `ColorWheel` class.

```
#include <QWidget>
#include <QString>
```

Classes

- class `ColorWheel`

The `ColorWheel` class.

8.4.1 Detailed Description

This file defines the `ColorWheel` class.

`colorwheel.h` and `colorwheel.cpp` are adapted from a github repository accesible at <https://github.com/liuyanghejerry/Qt-Plus/tree/master/develop/ColorWheel>

The author username is liuyanghejerry and he has given a universal concent to "use it in any purpose." This is locatable at: <http://qt-project.org/forums/viewthread/19757>

8.5 frame.cpp File Reference

```
#include "frame.h"
```

8.6 frame.h File Reference

This file defines the `Frame` class as well as the macros `FULLGRIDHEIGHT`, `FULLGRIDWIDTH`, `TOWERHEIGHT` and `TOWERWIDTH`.

```
#include "pixel.h"
#include <QtGlobal>
```

Classes

- class `Frame`

The `Frame` class.

Macros

- `#define FULLGRIDWIDTH 12`
Working area's width.
- `#define FULLGRIDHEIGHT 20`
Working area's height.
- `#define TOWERWIDTH 4`
Tower's width.
- `#define TOWERHEIGHT 10`
Tower's height.

8.6.1 Detailed Description

This file defines the [Frame](#) class as well as the macros FULLGRIDHEIGHT, FULLGRIDWIDTH, TOWERHEIGHT and TOWERWIDTH.

8.6.2 Macro Definition Documentation

8.6.2.1 `#define FULLGRIDHEIGHT 20`

Working area's height.

8.6.2.2 `#define FULLGRIDWIDTH 12`

Working area's width.

8.6.2.3 `#define TOWERHEIGHT 10`

Tower's height.

8.6.2.4 `#define TOWERWIDTH 4`

Tower's width.

8.7 main.cpp File Reference

```
#include "mainwindow.h"
#include "colorwheel.h"
#include <QApplication>
#include <QColor>
#include <QQmlComponent>
#include <QQmlEngine>
#include <QtDeclarative/QDeclarativeView>
```

Functions

- `int main (int argc, char *argv[])`

8.7.1 Function Documentation

8.7.1.1 `int main (int argc, char * argv[])`

8.8 mainwindow.cpp File Reference

```
#include "mainwindow.h"
```

8.9 mainwindow.h File Reference

This file defines the [MainWindow](#) class of the program.

```
#include <QMainWindow>
#include "ui_mainwindow.h"
#include "colorwheel.h"
#include "colorpalette.h"
#include "movie.h"
#include "timer.h"
#include "previewer.h"
#include <cstdlib>
#include <ctime>
#include <iostream>
#include <QtDeclarative/QDeclarativeView>
#include <QQtContext>
#include <QString>
#include <QPalette>
#include <QQuickItem>
#include <QFileDialog>
#include <QTime>
#include <QTimer>
#include <QMessageBox>
```

Classes

- class [MainWindow](#)
The [MainWindow](#) class.

Namespaces

- [Ui](#)

8.9.1 Detailed Description

This file defines the [MainWindow](#) class of the program.

8.10 mainwindow_util.cpp File Reference

```
#include "mainwindow.h"
```

8.11 movie.cpp File Reference

```
#include "movie.h"
```

Functions

- int [compareFrames](#) (const void *a, const void *b)
Used for the QSort function. It returns 1, 0, or -1 based on the timestamps of the two frames.

8.11.1 Function Documentation

8.11.1.1 int [compareFrames](#) (const void * a, const void * b)

Used for the QSort function. It returns 1, 0, or -1 based on the timestamps of the two frames.

8.12 movie.h File Reference

This file defines the [Movie](#) class.

```
#include <QList>
#include <QMediaPlayer>
#include "frame.h"
```

Classes

- class [Movie](#)
The [Movie](#) class.

Functions

- int [compareFrames](#) (const void *a, const void *b)
Used for the QSort function. It returns 1, 0, or -1 based on the timestamps of the two frames.

8.12.1 Detailed Description

This file defines the [Movie](#) class.

8.12.2 Function Documentation

8.12.2.1 int [compareFrames](#) (const void * a, const void * b)

Used for the QSort function. It returns 1, 0, or -1 based on the timestamps of the two frames.

8.13 pixel.cpp File Reference

```
#include "pixel.h"
```

8.14 pixel.h File Reference

This file defines the [Pixel](#) class.

```
#include <QColor>
#include <QPoint>
```

Classes

- class [Pixel](#)

The [Pixel](#) class.

8.14.1 Detailed Description

This file defines the [Pixel](#) class.

8.15 previewer.cpp File Reference

```
#include "previewer.h"
#include "ui_previewer.h"
```

8.16 previewer.h File Reference

This file defines the [Previewer](#) class.

```
#include <QDialog>
#include "movie.h"
#include "timer.h"
#include <QMediaPlayer>
#include <QQuickItem>
```

Classes

- class [Previewer](#)

The [Previewer](#) class.

Namespaces

- [Ui](#)

8.16.1 Detailed Description

This file defines the [Previewer](#) class.

8.17 timer.cpp File Reference

```
#include "timer.h"
```

8.18 timer.h File Reference

This file class defines the [Timer](#) class.

```
#include <QTime>
```

Classes

- class [Timer](#)
The [Timer](#) class.

8.18.1 Detailed Description

This file class defines the [Timer](#) class.

Index

- ~MainWindow
 - MainWindow, [22](#)
- ~Previewer
 - Previewer, [33](#)
- ~Timer
 - Timer, [35](#)
- ~colorPalette
 - colorPalette, [13](#)
- applyVector
 - Frame, [18](#)
- audio
 - MainWindow, [25](#)
- audioFile
 - Movie, [30](#)
- blue
 - MainWindow, [25](#)
- cWheel
 - MainWindow, [26](#)
- changeCurrentFrame
 - MainWindow, [22](#)
- closeEvent
 - MainWindow, [22](#)
- color
 - ColorWheel, [15](#)
 - Pixel, [32](#)
- colorChange
 - ColorWheel, [15](#)
- colorPalette, [13](#)
 - ~colorPalette, [13](#)
 - colorPalette, [13](#)
 - count, [14](#)
 - getColor, [13](#)
- colorPallet
 - MainWindow, [25](#)
- colorToString
 - ColorWheel, [15](#)
- ColorWheel, [14](#)
 - color, [15](#)
 - colorChange, [15](#)
 - colorToString, [15](#)
 - ColorWheel, [15](#)
 - composeWheel, [15](#)
 - current, [16](#)
 - drawIndicator, [15](#)
 - drawPicker, [15](#)
 - drawSquareImage, [15](#)
 - drawWheelImage, [16](#)
 - hueChanged, [16](#)
 - inSquare, [16](#)
 - inWheel, [16](#)
 - initSize, [16](#)
 - lastPos, [16](#)
 - margin, [16](#)
 - minimumSizeHint, [16](#)
 - mouseDown, [16](#)
 - mouseMoveEvent, [16](#)
 - mousePressEvent, [16](#)
 - mouseReleaseEvent, [16](#)
 - paintEvent, [16](#)
 - posColor, [16](#)
 - resizeEvent, [16](#)
 - setColor, [16](#)
 - sizeHint, [16](#)
 - squareImage, [16](#)
 - squareRegion, [16](#)
 - svChanged, [16](#)
 - updateColor, [16](#)
 - wheel, [16](#)
 - wheelImage, [16](#)
 - wheelRegion, [16](#)
 - wheelWidth, [16](#)
- colorpalette.cpp, [37](#)
- colorpalette.h, [37](#)
- colorwheel.cpp, [37](#)
- colorwheel.h, [38](#)
- compareFrames
 - movie.cpp, [41](#)
 - movie.h, [41](#)
- composeWheel
 - ColorWheel, [15](#)
- count
 - colorPalette, [14](#)
- current
 - ColorWheel, [16](#)
- currentAudio
 - Previewer, [34](#)
- currentFrameNumber
 - Movie, [30](#)
 - Previewer, [34](#)
- currentMovie
 - MainWindow, [26](#)
 - Previewer, [34](#)
- deleteCurrentFrame
 - Movie, [28](#)
- drawIndicator
 - ColorWheel, [15](#)

- drawPicker
 - ColorWheel, 15
- drawSquareImage
 - ColorWheel, 15
- drawWheelImage
 - ColorWheel, 16
- edited
 - MainWindow, 26
- FULLGRIDHEIGHT
 - frame.h, 39
- FULLGRIDWIDTH
 - frame.h, 39
- fileName
 - MainWindow, 26
- fixPalletBackground
 - MainWindow, 22
- Frame, 17
 - applyVector, 18
 - Frame, 18
 - fullGrid, 18
 - FullGridPixel, 18
 - getTimeStamp, 18
 - setFullGridPixelColor, 18
 - setTimeStamp, 18
 - setTowerGridPixel, 18
 - timeStamp, 18
 - towerGrid, 18
 - TowerGridPixel, 18
- frame.cpp, 38
- frame.h, 38
 - FULLGRIDHEIGHT, 39
 - FULLGRIDWIDTH, 39
 - TOWERHEIGHT, 39
 - TOWERWIDTH, 39
- frameSequence
 - Movie, 30
- fullGrid
 - Frame, 18
- FullGridPixel
 - Frame, 18
- fullTower
 - MainWindow, 26
- getAudioFile
 - Movie, 28
- getColor
 - colorPalette, 13
 - Pixel, 31
- getCurrentFrame
 - Movie, 29
- getFrame
 - Movie, 29
- getFrameCount
 - Movie, 29
- getFrameFromTime
 - Movie, 29
- getFrameNumber
 - Movie, 29
- getNextFrame
 - Movie, 29
- getTime
 - Timer, 35
- getTimeStamp
 - Frame, 18
- green
 - MainWindow, 26
- hueChanged
 - ColorWheel, 16
- inSquare
 - ColorWheel, 16
- inWheel
 - ColorWheel, 16
- initSize
 - ColorWheel, 16
- insertFrame
 - Movie, 29
- keyPressEvent
 - MainWindow, 22
- lastPos
 - ColorWheel, 16
- main
 - main.cpp, 40
- main.cpp, 39
 - main, 40
- mainFile
 - Movie, 30
- mainTower
 - MainWindow, 26
- MainWindow, 19
 - ~MainWindow, 22
 - audio, 25
 - blue, 25
 - cWheel, 26
 - changeCurrentFrame, 22
 - closeEvent, 22
 - colorPallet, 25
 - currentMovie, 26
 - edited, 26
 - fileName, 26
 - fixPalletBackground, 22
 - fullTower, 26
 - green, 26
 - keyPressEvent, 22
 - mainTower, 26
 - MainWindow, 22
 - on_SortButton_clicked, 24
 - on_actionAfter_triggered, 22
 - on_actionClear_Frame_triggered, 22
 - on_actionClose_triggered, 22
 - on_actionDelete_triggered, 22
 - on_actionExit_triggered, 22

- on_actionExport_triggered, 22
- on_actionNew_triggered, 23
- on_actionOpen_Audio_File_triggered, 23
- on_actionOpen_triggered, 23
- on_actionPreview_Mode_triggered, 23
- on_actionSave_As_triggered, 23
- on_actionSave_triggered, 23
- on_audioDurationChanged, 23
- on_blueSpinBox_valueChanged, 23
- on_currentTime_timeChanged, 23
- on_downButton_clicked, 23
- on_downLeftButton_clicked, 23
- on_downRightButton_clicked, 23
- on_greenSpinBox_valueChanged, 24
- on_leftButton_clicked, 24
- on_mediaSlider_sliderMoved, 24
- on_newFrameButton_clicked, 24
- on_playPauseButton_clicked, 24
- on_previewScrollBar_valueChanged, 24
- on_randomButton_clicked, 24
- on_redSpinBox_valueChanged, 24
- on_rightButton_clicked, 24
- on_stopButton_clicked, 24
- on_upButton_clicked, 24
- on_upLeftButton_clicked, 25
- on_upRightButton_clicked, 25
- ping, 26
- previewTowers, 26
- previewer, 26
- red, 26
- saveCurrentFrame, 25
- saveTimer, 26
- saveWarning, 25
- setUpMats, 25
- stop, 27
- timer, 27
- towerPainting, 25
- ui, 27
- updateColorSpinBoxes, 25
- updateMainTower, 25
- updateUI, 25
- mainwindow.cpp, 40
- mainwindow.h, 40
- mainwindow_util.cpp, 40
- margin
 - ColorWheel, 16
- minimumSizeHint
 - ColorWheel, 16
- mouseDown
 - ColorWheel, 16
- mouseMoveEvent
 - ColorWheel, 16
- mousePressEvent
 - ColorWheel, 16
- mouseReleaseEvent
 - ColorWheel, 16
- Movie, 27
 - audioFile, 30
 - currentFrameNumber, 30
 - deleteCurrentFrame, 28
 - frameSequence, 30
 - getAudioFile, 28
 - getCurrentFrame, 29
 - getFrame, 29
 - getFrameCount, 29
 - getFrameFromTime, 29
 - getFrameNumber, 29
 - getNextFrame, 29
 - insertFrame, 29
 - mainFile, 30
 - Movie, 28
 - newFrame, 29
 - printTimeStamps, 29
 - reset, 29
 - setAudio, 29
 - setFile, 30
 - setFrame, 30
 - setFrameNumber, 30
 - setFrameTime, 30
 - sortFrames, 30
- movie.cpp, 41
 - compareFrames, 41
- movie.h, 41
 - compareFrames, 41
- newFrame
 - Movie, 29
- offset
 - Timer, 36
- on_SortButton_clicked
 - MainWindow, 24
- on_actionAfter_triggered
 - MainWindow, 22
- on_actionClear_Frame_triggered
 - MainWindow, 22
- on_actionClose_triggered
 - MainWindow, 22
- on_actionDelete_triggered
 - MainWindow, 22
- on_actionExit_triggered
 - MainWindow, 22
- on_actionExport_triggered
 - MainWindow, 22
- on_actionNew_triggered
 - MainWindow, 23
- on_actionOpen_Audio_File_triggered
 - MainWindow, 23
- on_actionOpen_triggered
 - MainWindow, 23
- on_actionPreview_Mode_triggered
 - MainWindow, 23
- on_actionSave_As_triggered
 - MainWindow, 23
- on_actionSave_triggered
 - MainWindow, 23
- on_audioDurationChanged

- MainWindow, 23
- on_blueSpinBox_valueChanged
 - MainWindow, 23
- on_currentTime_timeChanged
 - MainWindow, 23
- on_downButton_clicked
 - MainWindow, 23
- on_downLeftButton_clicked
 - MainWindow, 23
- on_downRightButton_clicked
 - MainWindow, 23
- on_greenSpinBox_valueChanged
 - MainWindow, 24
- on_horizontalScrollBar_valueChanged
 - Previewer, 33
- on_leftButton_clicked
 - MainWindow, 24
- on_mediaSlider_sliderMoved
 - MainWindow, 24
- on_newFrameButton_clicked
 - MainWindow, 24
- on_playPauseButton_clicked
 - MainWindow, 24
- on_previewScrollBar_valueChanged
 - MainWindow, 24
- on_pushButton_clicked
 - Previewer, 33
- on_randomButton_clicked
 - MainWindow, 24
- on_redSpinBox_valueChanged
 - MainWindow, 24
- on_rightButton_clicked
 - MainWindow, 24
- on_stopButton_clicked
 - MainWindow, 24
- on_upButton_clicked
 - MainWindow, 24
- on_upLeftButton_clicked
 - MainWindow, 25
- on_upRightButton_clicked
 - MainWindow, 25
- operator=
 - Pixel, 31
- paintEvent
 - ColorWheel, 16
- ping
 - MainWindow, 26
- Pixel, 31
 - color, 32
 - getColor, 31
 - operator=, 31
 - Pixel, 31
 - setColor, 31, 32
- pixel.cpp, 41
- pixel.h, 42
- posColor
 - ColorWheel, 16
- previewTowers
 - MainWindow, 26
- Previewer, 32
 - ~Previewer, 33
 - currentAudio, 34
 - currentFrameNumber, 34
 - currentMovie, 34
 - on_horizontalScrollBar_valueChanged, 33
 - on_pushButton_clicked, 33
 - Previewer, 33
 - setUpMatrix, 33
 - showEvent, 34
 - stop, 34
 - timer, 34
 - tower, 34
 - ui, 34
 - updateData, 34
 - updateUI, 34
- previewer
 - MainWindow, 26
- previewer.cpp, 42
- previewer.h, 42
- printTimeStamps
 - Movie, 29
- red
 - MainWindow, 26
- reset
 - Movie, 29
- resizeEvent
 - ColorWheel, 16
- saveCurrentFrame
 - MainWindow, 25
- saveTimer
 - MainWindow, 26
- saveWarning
 - MainWindow, 25
- setAudio
 - Movie, 29
- setColor
 - ColorWheel, 16
 - Pixel, 31, 32
- setFile
 - Movie, 30
- setFrame
 - Movie, 30
- setFrameNumber
 - Movie, 30
- setFrameTime
 - Movie, 30
- setFullGridPixelColor
 - Frame, 18
- setTimeStamp
 - Frame, 18
- setTowerGridPixel
 - Frame, 18
- setUpMatrix
 - Previewer, 33
- setUpMats

- MainWindow, [25](#)
- showEvent
 - Previewer, [34](#)
- sizeHint
 - ColorWheel, [16](#)
- sortFrames
 - Movie, [30](#)
- squareImage
 - ColorWheel, [16](#)
- squareRegion
 - ColorWheel, [16](#)
- start
 - Timer, [35](#), [36](#)
- startTime
 - Timer, [36](#)
- stop
 - MainWindow, [27](#)
 - Previewer, [34](#)
 - Timer, [36](#)
- svChanged
 - ColorWheel, [16](#)
- TOWERHEIGHT
 - frame.h, [39](#)
- TOWERWIDTH
 - frame.h, [39](#)
- timeStamp
 - Frame, [18](#)
- Timer, [35](#)
 - ~Timer, [35](#)
 - getTime, [35](#)
 - offset, [36](#)
 - start, [35](#), [36](#)
 - startTime, [36](#)
 - stop, [36](#)
 - Timer, [35](#)
 - timer, [36](#)
- timer
 - MainWindow, [27](#)
 - Previewer, [34](#)
 - Timer, [36](#)
- timer.cpp, [43](#)
- timer.h, [43](#)
- tower
 - Previewer, [34](#)
- towerGrid
 - Frame, [18](#)
- TowerGridPixel
 - Frame, [18](#)
- towerPainting
 - MainWindow, [25](#)
- Ui, [11](#)
- ui
 - MainWindow, [27](#)
 - Previewer, [34](#)
- updateColor
 - ColorWheel, [16](#)
- updateColorSpinBoxes
 - MainWindow, [25](#)
 - Previewer, [34](#)
- updateData
 - Previewer, [34](#)
- updateMainTower
 - MainWindow, [25](#)
- updateUI
 - MainWindow, [25](#)
 - Previewer, [34](#)
- wheel
 - ColorWheel, [16](#)
- wheelImage
 - ColorWheel, [16](#)
- wheelRegion
 - ColorWheel, [16](#)
- wheelWidth
 - ColorWheel, [16](#)